

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-288561

(43)Date of publication of application : 10.10.2003

(51)Int.Cl.

G06K 19/07

G06K 17/00

(21)Application number : 2002-092902 (71)Applicant : TOPPAN PRINTING CO LTD

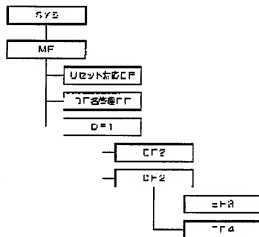
(22)Date of filing : 28.03.2002 (72)Inventor : NOZAKI NAOTO

(54) IC CARD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an IC card which enables initial response data to speedily be updated if the initial response data, registered in a ROM, etc., as a memory whose storage contents are unrewritable, need to be altered afterward.

SOLUTION: In a nonvolatile memory, a reset response EF for writing new initial response data is provided and when a reset signal is received from outside, it is decided whether the new initial response data are written in the reset response EF; when the new initial response data are written in the reset response EF, the initial response data written in the ROM and the new initial response data in the reset response EF are read out and outputted to the outside.



LEGAL STATUS

[Date of request for examination] 13.01.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's
decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

(51) Int.Cl. ⁷	識別記号	F I	テコード(参考)
G 0 6 K	19/07	G 0 6 K	D 5 B 0 3 5
	17/00		N 5 B 0 5 8
		19/00	

審査請求 未請求 請求項の数 2 O L (全 29 頁)

(21) 出願番号 特願2002-92902X(P2002-92902)

(22) 出願日 平成14年3月28日 (2002. 3. 28)

(71) 出願人 000003193

凸版印刷株式会社

東京都台東区台東1丁目5番1号

(72) 発明者 野崎 直人

東京都台東区台東1丁目5番1号 凸版印

刷株式会社内

(74) 代理人 100064908

弁理士 志賀 正武 (外6名)

Fターム(参考) 5B035 A400 B403 B809 CA01 CA04

CA08 CA11 CA22 CA29

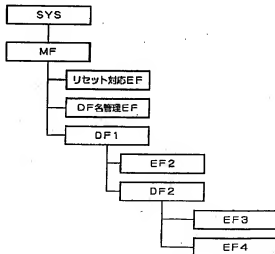
5B058 CA12 CA23 KA01 KA11

(54) 【発明の名称】 I Cカード

(57) 【要約】

【課題】 記憶内容を書き換え不可能なメモリであるROM等に登録した初期応答データを、後に変更する必要がある場合には、速やかに初期応答データを更新することが可能なI Cカードを提供する。

【解決手段】 不揮発性メモリ内に、新たな初期応答データを書き込むためのリセット応答EFを設け、外部からリセット信号を受信した場合に、リセット応答EF内に新たな初期応答データが書き込まれているか否かを判断し、リセット応答EF内に新たな初期応答データが書き込まれていた場合には、ROM内に書き込まれている初期応答データとリセット応答EF内の新たな初期応答データとを読み出して、外部に出力する。



【特許請求の範囲】

【請求項1】 記憶内容を書き換え可能な不揮発性メモリと、該不揮発性メモリを制御する中央処理装置と、当該カード固有のプロトコルを外部に通知するための初期応答データが予め格納されているROMとを有するICカードであって、

前記不揮発性メモリ内に、新たな初期応答データを書き込むための初期応答データ書き込み専用ファイルを具備することを特徴とするICカード。

【請求項2】 外部からリセット信号を受信した場合に、前記初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれているか否かを判断し、前記初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれていた場合には、前記ROM内に書き込まれている初期応答データと前記初期応答データ書き込み専用ファイル内の新たな初期応答データとを読み出して、外部に出力することを特徴とする請求項1に記載のICカード。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、不揮発性メモリを搭載したICカードに係り、特にATR (Answer To Reset) 情報を更新することが可能なICカードに関する。

【0002】

【従来技術】半導体メモリ等を内蔵するICカードの登場により、従来の磁気カード等に比べて記憶容量が飛躍的に増大するとともに、マイクロコンピュータ等の半導体集積回路装置を内蔵することによってICカード自体が演算処理機能を有することで情報媒体に高いセキュリティ性を付与することができるようになった。

【0003】ICカードはISO (International Organisation for Standardisation) で国際的に規格化されており、一般的にICカードはプラスチックなどを基材としてカード本体に半導体メモリ等のICが内蔵され、カード表面に外部装置との接続のために金属製の導電性端子が設けられており、そのICカードと外部装置とのデータの交換のためにICカードを外部装置のカードスロットに挿入して用いるものである。このICカードは接触型ICカードと呼ばれ、特に、大量データ交換や決済業務等通信の確実性と安全性が求められる用途、例えばクレジットや電子財布等に好適である。

【0004】一方、入退室等のゲート管理への適用に際しては、認証が主たる通信内容であって、通信データも少量の場合が多く、より簡略な処理が望まれる。この問題を解決するために考案された技術が非接触型ICカードである。これは、空間に高周波電磁界や超音波、光等の振動エネルギーの場を設けて、そのエネルギーを吸収、整流してカードに内蔵された電子回路を駆動する直流電力源とし、この場の交流成分の周波数をそのまま用いるか、或いは通信や分周して識別信号とし、この識別

信号をコイルやコンデンサ等の複合器を介してデータを半導体素子の情報処理回路に伝送するものである。

【0005】特に、認証や単純な計数データ処理を目的とした非接触型ICカードの多くは、電池とCPU (Central Processing Unit ; 中央処理装置) を搭載しないハードロジックの無線認証 (いわゆる、Radio Frequency Identification) であり、この非接触型ICカードの出現によって、磁気カードに比較して偽造や改竄に対する安全性が高まった。また、この非接触カードによれば、無線によって外部装置と通信を行うことができるため、例えば、駅の改札等に代表されるようなゲートを通る場合には、カード携帯者は、携帯するICカードをゲートに設けられているICカードリーダー (以下、カードR/Wと記載する) に接近させるだけで良く、従来のゲート通過に伴う煩わしい作業をすることなく、簡単にゲートを通ることができる。

【0006】また、近年になって、多目的な用途に1枚のカードで対応することを目的として前者の外部端子を持つ接触型の機能と後者の無線通信によってデータ通信する非接触型の機能を有する複合型のICカードが考案されている。

【0007】

【発明が解決しようとする課題】上述したようなICカードは、通常、外部装置であるカードR/Wとの間でデータの送受を行う場合、カードR/Wからのリセット信号により制御プログラムが起動され、その後、カードR/Wからのコマンドを受け取って解釈し、これに対応する処理を実行した後、その処理結果を応答データとしてカードR/Wに出力し、その後待機状態となる。ところで、一般に、ICカードは機種によりその伝送プロトコルが異なる。このため、ICカードは伝送プロトコル等をカードR/Wへ通知するため、リセット信号によるリセット解除を認識後、ATR (Answer To Reset) 情報と称する初期応答データをカードR/Wに出力する。

【0008】上記ATR情報は、当該ICカードの伝送プロトコル仕様情報とICカード固有の情報とから構成されており、通常、ICカードに搭載されるICチップの製造工程で、ICチップ内のマスクROM内に書き込まれる。ここで、上記マスクROMは、書き換え不可能なメモリであるため、マスクROMに書き込んだATR情報を後に書き換えることは不可能である。

【0009】しかしながら、ICカード発行後にいて、ICカードのソフトウェアをバージョンアップした場合や、ICカードに新たなソフトウェアを追加した場合等には、ROM内に登録されている上記ATR情報を更新する必要がある。このようなATR情報の変更が必要となる場面は、現在多く見受けられるが、従来のICカードにおいては、ROM内に書き込まれているATR情報を書き換える又は追加することができなかったため、ICカード製造後に生じるデータ更新等に対応す

ることができなかった。

【0010】本発明はこのような事情に鑑みてなされたもので、記憶内容を書き換え不可能なメモリであるROM等に登録した初期応答データを、後に変更する必要がある場合には、速やかに初期応答データを更新することが可能なICカードを提供することを目的とする。

【0011】

【課題を解決するための手段】上記目的を達成するために、本発明は、記憶内容を書き換え可能な不揮発性メモリと、該不揮発性メモリを制御する中央処理装置と、当該カード固有のプロトコルを外部に通知するための初期応答データが予め格納されているROMとを有するICカードであって、前記不揮発性メモリ内に、新たな初期応答データを書き込むための初期応答データ書き込み専用ファイルを提供することを特徴とするICカードを提供する。

【0012】また、請求項2に記載の発明は、請求項1に記載のICカードにおいて、外部からリセット信号を受信した場合に、前記初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれているか否かを判断し、前記初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれていた場合には、前記ROM内に書き込まれている初期応答データと前記初期応答データ書き込み専用ファイル内の新たな初期応答データとを読み出して、外部に出力することを特徴とする。

【0013】なお、本発明のICカードの用途の代表的なものとしては、例えば、金融関係におけるキャッシュカード、クレジットカード、通帳、行政関係における印鑑証明カード、住民基本台帳カード、年金手帳、運転免許証、パスポート、資格保有の証明書が例として挙げられる。また、有価証券関係における商品券、プリペイドカード、株券として、運輸関係における定期乗車券、回数券、ETCカード、運賃精算券、有料道路通行券等が例として挙げられる。また、医療関係においては、健康保険証、診療券、母子手帳、カルテ、診療履歴証、医療個人証（血液型、投与禁止薬情報等を入力）、献血手帳等が、また、ID関係、その他の分野においては、社員証、会員証、保険証券、ポイントカード、身分証明書、学生証、成績表、給与明細表、出勤管理カード、電子鍵、入退室管理カード、図書館利用証、機器利用証、施設利用証、食堂利用証、各種決済カード、電話帳、警察手帳等が例として挙げられる。

【0014】

【発明の実施の形態】以下、図面を参照し、本発明の実施形態について説明する。まず、接触型、非接触型、複合型等の各種ICカードについて、それぞれ基本構造、基本動作等について概要を説明する。

【0015】（1-1）接触型ICカードの概要

代表的な接触型ICカードの概観図を図20に、接触型

ICカードの電気回路構成図を図21に示す。図20に示すように、接触型ICカード100は、プラスチックカード本体の表面の一部に外部装置とのインターフェースとなる端子電極101が取り付けられ、その端子電極101の下面にICチップ102が取り付けられている。また、カード本体の表面には、印刷文字、エンボス文字103の形成が可能である。また、既存の磁気カードシステムにも対応可能とするため、磁気ストライプ104が設けられている場合もある。上記ICチップ102の電気回路構成は、図21に示すように、CPU（中央処理装置）105とメモリとを備えている。メモリは、ROM（Read Only Memory）106、RAM（Random Access Memory、揮発性メモリ）107、記憶内容を書き換え可能な不揮発性メモリとしてのEEPROM（Electrically Erasable Programmable ROM）108等からなる。上記ROM106にはCPU105が実行するICカード動作プログラムを格納されている。また、RAM107は、コマンド（命令）等のデータの一時格納領域やCPU105の作業領域等として使用される。EEPROM108には、メモリデータや各種設定等が格納される。CPUは、これらのメモリからデータを読み出すため、また、書き込みを行うために所定幅のバスを介して接続されている。

【0016】また、外部装置とのインターフェースである端子電極101には、外部より電源を供給するための電源端子Vcc及びグラウンド端子GNDと、外部よりリセット信号を受信するためのリセット信号端子RSTと、外部との間でデータの送受信するためのI/O端子と、外部よりシステムクロック信号を受信するためのクロック端子CLKとを備えている。

【0017】また、CPU105は、外部装置から供給されるVcc（電源電圧）、RST信号（リセット信号）、CLK信号（クロック信号）に基づいて、ROM106に格納されたプログラムを読み出して実行し、RAM107およびEEPROM108に対してデータの書き込み及び読み出しを行うとともに、I/O端子を介して、外部機器からのコマンド、書き込みデータ等を受信し、また、ICカードからのレスポンスとしての処理結果・読み出しデータ等を送信する。接触型ICカード100は、記憶容量が大きいため、ソフトウェアも搭載可能であり、このため多機能な1枚のICカードで実現することが可能である。また、セキュリティ性に優れ、偽造、不正使用等が行われにくいという特徴を有している。

【0018】（1-2）非接触型ICカードの概要
以下に、非接触型ICカードの基本的構成と基本原理について図面を用いて説明する。図22に示すように、非接触型ICカード109は、図中点線で示すようにカード本体内部にICチップ110、コイル111等を実装し、無線により外部装置とデータの送受を行う。このた

め、ICカード表面には、上述した接触型ICカード100のように端子電極101を要しない。次に、図23に非接触型ICカード109の電気回路構成を示す。同図に示すように、非接触型ICカード109は、上記ICチップ110と、共振回路部116とを備える。上記ICチップ110には、CPU(中央処理装置)111とROM112、RAM113、EEPROM114及びRF回路115が実装されている。なお、ICチップ110の電源安定のため、バイパスコンデンサ(図省略)が設けられる場合もある。

【0019】上記共振回路部116は、コイル117とコンデンサ118とによって構成され、その共振周波数が外部装置120の放射する高周波電磁界の周波数と等しくなるように設定されている。これにより、外部装置120の送受信アンテナと直接電磁的に結合され電力の受信や情報の送受を可能とする。尚、ICチップは各社が開発・設計しており、ICチップの製品設計によっては、非接触型ICカードに実装する際に、必ずしも上述したような共振周波数を調整する工程を必要としない例もあり、もちろんその場合には、このように共振周波数を調整するための電気部品を設けるなどの工程は必要ない。

【0020】ここで、外部装置120から非接触型ICカード109に電力および情報を伝達する場合における、コイル117の動作を以下に説明する。外部装置120の送受信回路で発生された図示しない高周波信号により、外部装置120内部の送受信コイルに高周波電磁界が誘起され、この高周波信号が、電磁エネルギーとして空間に放射される。このとき、非接触型ICカード109がこの高周波電磁界中に位置すると、外部装置120の送受信コイルにより発生された高周波電磁界により、電磁誘導によって、非接触型ICカード内の共振回路部116に高周波電流が流れる。これにより、非接触型ICカード109は起電力を得る。ここで、コイル117とコンデンサ118による共振回路部116の共振周波数は、外部装置120が放射する高周波電磁界の周波数に鋭く共振するように設定されている。これにより、外部装置120から最大の電磁エネルギーを受けることができる。また、共振回路部116は、閉ループを形成しているため受信したエネルギーを蓄積エネルギーとして保有することができる。

【0021】なお、コイル117の形成方法として、銅箔やアルミ箔のエッチングにより形成したり、銀ペーストを印刷したり、電線巻いたりする方法が知られており、また、コンデンサ118の形成方法として、アンテナをフレキシブル基板としてその基板の両面に電極をつけてコンデンサを形成する方法や、フィルムコンデンサやセラミックコンデンサを実装する方法が知られている。なお、コイル117及びコンデンサ118は、図23に示すようにICチップ110の外部に形成される

場合の他、ICチップ110内に形成される技術もある。なお上述した技術は、後述の複合型ICカードに適用することも可能である。

【0022】また、非接触型ICカード109の製造方法としては、一般的にラミネート技術によるものが知られている。このラミネート技術では、プレス製の2枚の板の間に複数の熱可塑性シートを積み重ねて配置し、非接触型のICチップ110をその中間に配置する。このICチップ110は、予めこのICチップ110を取り囲むコイル116等と電氣的に接続されている。その配置後、熱と圧力を加えて複数の熱可塑性シートを溶着させ、ICチップ110と熱可塑性シートを一体化させ、非接触型ICカード109を製造する。また、上記ラミネート技術によるもの他、例えば、熱可塑性シートの間にカードのサイズに合った矩形のフレームを設置し、このフレームと中間のシートによって形成された空洞内に、予めコイル117に接続されているICチップ110を設置し、上記空洞に熱硬化樹脂を注入し、その後、この空洞を上記熱可塑性シートで被覆するという製造方法もある。なお、アンテナ(一般的にコイル状のものが多い)とICチップとの配置は、上述した例に限らず、適宜設計しても良い。また、アンテナとICチップとの間は、必ずしも導体で接続される必要はなく、例えば、トランス結合などの電気工学技術を利用して、アンテナとICチップとの間のデータ送受信などを行うようにしてもよい。

【0023】上述した非接触型ICカード109は、汚れ、埃、始動等の外界からの影響を受けにくく、また、操作性、端末のメンテナンスに優れる等の特徴を有する。また、外部装置120と接触さなくても情報の授受が可能であることから、利便性、保守性に優れ、例えば、鉄道等の乗車券として有効に活用することが可能である。

【0024】(1-3) 複合型ICカードの概要
複合型ICカードは、上述した接触型、非接触型を一体化したものである。複合型ICカードと称する他、ハイブリッド型ICカード、コンビ型ICカード、コンビカード、というように呼ばれることもある。また、複合型ICカードは、インターフェースのタイプ等によって、デュアルインターフェースタイプと統合タイプに分かれる。

【0025】デュアルインターフェースタイプは、接触型、非接触型のインターフェースを有するが、CPU、メモリ(ROM、RAM、EEPROM等)を共用するタイプである。例えば、1つのICチップに接触型、非接触型の2種類のインターフェース機能とOS(オペレーティングシステム)を備え、一つのCPUでメモリを共用する。なお、デュアルインターフェースICカード、デュアルICカード、デュアルカードというように呼ばれることもある。これに対し、統合タイプは、接触型の

ICチップ、非接触型のICチップの、二つのICチップを有し、CPU、メモリを共用しないタイプである。

【0026】複合型ICカードは、上述した接触型と非接触型との二つの機能を有するため、用途に応じて広く使い分けることが可能である。即ち、接触型ICカードが好ましいとされるクレジットカード、キャッシュカード、物販、認証利用等の高価決済、認証サービス、端末アクセス管理、ポイントサービス、非接触型ICカードが好ましいとされる物販飲食利用の少額決済、入退室管理、ゲート利用、出欠席管理、出勤勤管理、ポイントサービスカード、定期券、乗車券等に有効に活用することができる。

【0027】続いて、代表的なデュアルインターフェースタイプの複合型ICカード220(以下、単に複合型ICカードと記載する)の電気回路構成を図24に示す。図面に示すように、複合型ICカード220は、ICカード全体の制御を行うCPU121と、ブートプログラム等の固定データが予め記憶されたROM122と、電気的にデータの書き込み、消去が可能な不揮発性メモリであるEEPROM123と、各種データを一時的に記憶するRAM124と、外部装置の端子電極と接触することにより、電力、データの送受を行う接触型インターフェース(I/F)としての端子電極125と、非接触型インターフェース(I/F)としての共振回路部126と、共振回路部126とCPU121との間に介在するRF回路127とを備えている。

【0028】接触型インターフェースとして機能する上記端子電極125は、外部より電源を供給するための電源端子Vcc及びグラウンド端子GNDと、外部よりリセット信号を受信するためのリセット信号端子RSTと、外部との間でデータの送受信するためのI/O端子と、外部よりシステムクロック信号を受信するためのクロック端子CLKとを備えている。上述した金属製の各端子は、ICカードの表面に配置されており、これらの端子を外部装置である接触型カードR/Wに電気的に接触されることにより、電力の供給やデータの送受を行う。

【0029】一方、非接触型インターフェースとして機能する共振回路部126は、図23に示した上述の非接触型ICカードと同様に、ICカード内部に形成されたコイル状のアンテナ及びコンデンサとから構成され、外部装置である非接触型カードR/Wと電波により電力の供給やデータの送受を可能とする。共振回路部126によれば、ICカード本体がカードR/Wに接触しないため、摩擦部分が少なく、カードR/W及びICカードの双方の寿命が長くなる他、アンテナがICカードの内部に設置されているため、カードの表面と裏面とを全て印刷して使用できる、複数枚のICカードに対してほぼ同時に読み書きができる等の利点がある。

【0030】また、上記ROM122には、接触型イン

ターフェースに対応するオペレーティングシステムと、非接触型インターフェースに対応するオペレーションシステムとがそれぞれ記憶されている。また、EEPROM123には、接触型オペレーションシステム又は非接触型オペレーションシステム上で動作するアプリケーション、認証キー情報、データ等が格納されている。

【0031】次に、図25に上記複合型ICカード220の構造を示す。図25(a)は、全体構造を示しており、図25(b)は、複合型ICモジュール実装部を横切る横断面図を示したものである。図面において、共振回路部126を構成するコイル128は、シート状の樹脂の表面にプリントパターンで形成されており、上記コンデンサ129は、シート状樹脂の誘電体を介して表面・裏面それぞれ対向した平行平板からなる。

【0032】複合型ICモジュール130は、複合型ICチップ131と、モジュール基板132と、端子電極125とからなる。上記複合型ICチップ131は、図24に示したCPU121、ROM122、EEPROM123、RAM124、及びRF回路127が内蔵されたICチップである。複合型ICチップ131と端子電極125とは、モジュール基板132の裏面に実装され、スルーホールで互いに回路接続された後に、複合型ICチップ131は樹脂封止される。これにより、複合型ICモジュール130が完成する。

【0033】続いて、樹脂基板に、プリントパターンによってコイル128と裏表面对向した平行平板のパターンによるコンデンサ129からなる共振回路部126を形成し、コイルとして機能するフレキシブルなアンテナ基板134が準備される。ここでは、コイル128はプリントパターンによる形成としたが絶縁被覆した導線を巻いて形成する方法でもよい。また、アンテナ基板134の樹脂としては塩化ビニルの他、ポリイミド、ポリカーボネート、PET等が適用でき、材料は一種に固定されるものではない。

【0034】続いて、射出成形によりアンテナ基板134を封入してカード基板135を作製する。成形の際、アンテナ基板134は共振回路部126と複合型ICモジュール130の実装位置とが、所定の位置となるように配置される。また、射出成形によるカード基板135の製作と同時に、表面となるカード基板135には、複合型ICモジュール130の嵌合穴136を形成する。最後に、カード基板135の複合型ICモジュールの嵌合穴136に複合型ICモジュール130を導電ペースト等で接着することにより、複合型ICカード220が完成する。なお、カード基材としては塩化ビニルの他、ポリカーボネートなど十分な強度とエンボス性などカードの特性が得られるものであれば必ずしも本発明に適用できる。

【0035】なお、図25(d)では、カード基板135は、表面と裏面とで変色して描いてあるが、本来、一体

のものであり、ここでは、カード基板 135 に封入されるアンテナ基板 134 における共振回路部 126 と、複合型 IC モジュール 130 が取り付けられる嵌合穴 136 との関係を確認するために修飾したものである。また、カードの製作は、上述した射出成形の他、エンボス特性を維持する方法であれどいずれも適用可能であり、例えば、ラミネート方式、接着剤充填方式等が用いられる。また、複合型 IC モジュール 130 の嵌合穴 136 を、カード成形後にくりぬき加工することもある。

【0036】続いて、エンボスに対応した複合型 IC カード 220 の平面図を図 26 に示す。この例では、コイル 128 を外部端子領域 137 に実装される複合型 IC モジュール 130 に近接させ、更に外部端子領域 137 (複合型 IC モジュール 130) に該コイル 128 のループの外側に位置するように配置して、エンボス領域 138・外部端子領域 137 を干渉しないように配置した場合を示している。これはコイル仕様が所定のインダクタンスを得るために、コイル 128 がスパイラル形状であり、そのコイル 128 の幅がエンボス領域 138 と外部端子領域 137 との間隔よりも広くなった場合に非常に有効である。

【0037】次に、図 27 (a)、(b) にエンボス領域 138 に対応した複合型 IC カード若しくはエンボス領域 138 無しの場合の複合型 IC カードにおけるコイル 128 の複合型 IC カード 220 内部に於ける実装位置を示す。コイル 128 はカードのほぼ周囲全体に配置されている。このときのコイル仕様は、カードのほぼ全周に沿って配置されるため、所定のインダクタンスを得るのに図 26 における場合と比較して少ない巻数で実現できる。また、コイル 128 の外部端子領域 137 に近接する長さは任意であり、図 27 (b) に示すように、外部端子領域 137 のほぼ 3/4 周を取り囲むように配置することも可能である。この場合におけるアンテナ基板 134 はエンボス領域 138 に対応する部分の樹脂シートを切り抜いてある。これは、エンボス特性に影響を与えないことを目的としている。

【0038】また、図 26 及び図 27 では、コイル 128 の巻き方 (ループ状態) が比較的均等に揃っていたが、図 28 に示すように、コイル 128 の巻き数を非常に多くし、外周径と内周径の差を大きくするようにしてもよい。図 28 では、例えば、外部装置が放射する高周波電磁界の周波数が、図 26 及び図 27 の場合に比べて 1 桁乃至 2 桁低い場合に適したコイルの巻き方である。高周波電磁界の周波数が低くなると、当然のことながら共振回路部 126 でのコイル 128 のインダクタンス値およびコンデンサ 129 の容量値が増加する。コイル 128 のインダクタンス増加は巻数の増大につながり、プリントパターンによるスパイラルコイル形状では外周径と内周径の差が大きくなり図 28 に示す形態となる。

【0039】(2) IC カードシステムの概要

次に、上述したような IC カードと通信を行うシステム側 (外部装置側) の構成について図 42 を参照して説明する。

(2-1) IC カードシステムの構成

図 29 に示すように、IC カードシステムは、一般にホストコンピュータ 140、端末装置 141、及びカード R/W 142 から構成される。ホストコンピュータ 140 は、IC カードシステム全体を集中的に制御するものである。端末装置 141 は、ホストコンピュータ 140 に通信回線などで接続されており、各地に分散して多数配置してある。カード R/W 142 は、例えば各店舗に設置され、IC カードとホストコンピュータ 140 側とのインターフェースとして機能する。なお、ホストコンピュータ 140 には、カード発行装置 143 が接続されることもある。また、上記端末装置 141 としては、専用に作られた専用端末の他に、パソコン、キャッシュペーパー端末、ATM 端末、自動販売機、入退場ゲート、POS (Point Of Sales) 端末、携帯電話が例として挙げられる。

【0040】図 29 に示した IC カードシステムにおいて、接触型のカード R/W (IC カードとしては接触型、複合型) が使用される場合は、図 30 に示すように、カード R/W 142 に IC カード 144 が挿入されることにより、カード R/W 142 の端子電極と 145 と、IC カード 144 の端子電極 146 とが接触し、この端子電極を介して信号の送受信がなされる。具体的にはカード R/W 142 から IC カード 144 へ制御信号が送信され、これに対する応答信号が IC カード 144 からカード R/W 142 へ送信される。また、IC カードに情報を書き込む場合には、カード R/W 142 と IC カードとの間でネゴシエーションを行い、カード R/W 142 が書き込みの許可を得た場合、書き込むデータを IC カード 144 へ送信することで、該データの書き込みを行うことができる。

【0041】一方、IC カード 144 から情報を読み出す場合には、カード R/W 142 は、IC カード 144 に対して読み出し要求と読み出しを所望するファイルの情報を送信することにより、所望の情報を IC カード 144 から読み出すことが可能となる。

【0042】また、図 29 に示した IC カードシステムにおいて、非接触型のカード R/W (IC カードとしては非接触型、複合型) が使用される場合は、図 31 に示すように、IC カード 148 をカード R/W 147 に接近させることにより、電磁誘導によりカード R/W 147 から起電力を得、カード R/W 147 との間で無線により上述したような通信を行うことができる。

【0043】(3) IC カードの製造から発行まで
続いて、IC カードの製造から発行までの過程を図 32 を参照して順に説明する。まず、IC カードの製造過程

(図32のステップSP41)では、ICカード用ICチップのOS、アプリケーションソフトの開発、ICチップの設計工程、ICチップの製造工程、ICチップをプラスチック等のカードに実装する工程、ICカードに文字や模様等の印刷を行なう工程を要する。そして、このようにして製造されたICカードは、発行者に渡され、発行処理が行なわれる(ステップSP42)。

【0044】発行処理では、カードのフォーマット(EEPROMの初期化処理)、CDF(Common Data File)を生成する処理が行われる。また、CDFには、カードの属性情報、カードの発行者、カードの所有者を特定する情報、アプリケーションソフトへのアクセス権限の情報、PIN(Personal Identification Number)情報(個人識別情報)等が記憶される。また、ADF(Application Data File)を生成する。ADFには、アプリケーション内のファイルへのアクセス権限の情報、アプリケーション内で用いられるPIN情報等が格納される。

【0045】そして、上述したような発行処理されたICカードは、使用者に渡され、サービス提供者が提供するサービスを、使用者がICカードを使って受けることができる(ステップSP43)。なお、初回使用時に、使用者が端末を用いて暗証番号等を設定する場合もある。

【0046】(4) ICカードのメモリ管理方法
次に、ICカードのメモリ管理方法の一実施形態について図1～図15を参照して説明する。なお、ここでのメモリ管理方法は、上述した接触型、非接触型、複合型のICカードにそれぞれ適用可能であるが、簡単のため、図20～図21において説明した接触型ICカード100に本発明のメモリ管理方法を適用する場合について説明する。

【0047】まず、図33に、原国際規格ISO/IEC 7816シリーズにより規定されている基本的なファイル階層構造を示す。このファイル階層において、MF(Master File)は最上位に位置するファイルであり、ファイル構成の根幹となる唯一の必須専用ファイルである。そして、上記MFの配下に、DF(Dedicated File)、及びEF(Elementary File)が構成される。DFは、ファイル制御情報及び任意選択の付加可能メモリを含むファイルであり、EF、及び又は専用ファイルの親ファイルとなることができる。EFは、同一のファイル識別子を有しているデータ単位又はレコードの集合であり、アプリケーションプログラム、データ等を格納するWEF(Working EF)と、認証鍵及び暗号を格納するIEF(Internal EF)等がある。そして、上述したような階層構造を持つファイルが上述した各種ICカードに使用される不揮発性メモリとしてのEEPROMに格納されている。

【0048】(4-1) 本実施形態に係るファイル階層

構造

続いて、本実施形態における具体的なファイル階層構造を図1に示す。なお、本実施形態では、図20に示した接触型ICカード100のEEPROM108に図1に示したファイルが格納される。

【0049】このファイル階層構造は、図33に示した原国際規格ISO/IEC 7816シリーズに準拠したものであり、その構造は、ファイルの最上位に配置されたMFの配下に、リセット応答EF、DF名管理EF、DF1が連なり、更に、DF1の配下に、EF2及びDF2が連なり、DF2の配下にEF3及びEF4が連なる階層構造となっている。なお、上記MFの上位には、当該MF並びにMF配下に存在するファイルを格納するEEPROMのメモリマップの状態を管理するシステムディレクトリsysが生成される。

【0050】(4-2) 本実施形態に係るディレクトリ及びファイルのレイアウト

そして、各ファイルは、ディレクトリによって管理される。ディレクトリは、図2に示すようなレイアウトを有し、当該ディレクトリが生成されているものであるか又は削除されたものであるかを示すディレクトリ生成・削除フラグFLG、ファイルID(ファイル番号)FID、配下に存在するファイルの先頭セクタ(区画)番号(先頭論理アドレス)TOP、配下に存在するファイルの最終セクタ(区画)番号(最終論理アドレス)BTM、ファイルサイズ(全セクタ数)VLMをそれぞれ登録するメモリ領域を有する。また、上述したDFには、ファイル名を付与することができ、ファイル名を書き込むメモリ領域のレイアウトは、図3に示すように、ディレクトリ生成・削除フラグFLG、DFディレクトリの先頭論理アドレスTOP、及びファイルのID名(DF名)を格納するメモリ領域とを有する。

【0051】なお、本実施形態において、ファイルの生成、削除は、32バイトからなる1セクタ(1区画)を最小単位として取り扱う。また、上記ディレクトリは原則として1セクタを要する。

【0052】(4-3) ファイルの生成
以下、図4に示したファイル階層構造と、図5～図15に示したEEPROMのメモリマップの遷移図を参照し、ファイル生成の処理について説明する。なお、図4は、図1に示したファイル階層において、ディレクトリを構造内に表したものである。

【0053】まず、EEPROM(不揮発性メモリ)の初期状態におけるメモリマップを図5に示す。図5に示すように、EEPROMのメモリ領域は、ファイル領域と、FAT(file Allocation Table)領域(ファイル管理領域)とに分割されている。本実施形態では、物理アドレス「H'000000」～「H'049990」をファイル領域に、「H'050000」以降の物理アドレスをFAT領域に割り当てられている。

【0054】上記ファイル領域においては、32バイトからなる1セクタ毎に論理アドレスが割り当てられている。なお、図中「H」で始まるアドレスは物理アドレスを示しており、「H」で始まるアドレスは、論理アドレスを示している。ここで、論理アドレスとは、EEPROM上に実際に割り当てられたアドレスではなく、ファイルを管理するためにプログラム上で使用される仮想（論理上）のアドレスであり、16進で表記されている。これに対し、物理アドレスとは、1バイト毎に割り当てられている番地であり、16進で表記されている。なお、ハードウェアでは、物理アドレスを使用しているため、ソフトウェア上において、物理アドレスと論理アドレスとの変換処理が必要となる。

【0055】一方、FAT領域には、セクタ（区画）Yの各々に対応する管理セクタFを設け、管理セクタFの各々には、対応する区画の使用状態等を登録する。互いに対応するセクタYと管理セクタFとは、同一の論理アドレスで識別され、図中網掛けで示されている番号は、FAT領域において各管理セクタFに割り当てられた論理アドレスである。

【0056】なお、FAT領域における論理アドレス（FAT領域における網掛け部分）は、実際には、メモリマップ上には存在しない。また、FAT領域における管理セクタFは2バイトで構成されているため、2バイト毎に論理アドレスが割り当てられていることとなる。なお、FAT領域においては、1行16バイトで表示しているため、2行で1セクタYに相当する32バイトを示している。

【0057】また、図5においては、全てのメモリエリアに初期値「FFFF」が記憶されている。この状態は、EEPROMの全てのメモリ領域にデータが何も入っていない状態を表しており、フォーマット前、即ち、製造した直後のEEPROMの状態に相当する。なお、「FFFF」に代わって「0000」が記憶されている場合もある。

【0058】続いて、接触型ICカード100が外部装置（例えば、図43におけるカードR/W142）に挿入されることにより、I/O端子を介してフォーマット実行のコマンドと、確保するセクタ数とを受信すると、ICカード内のCPU105は、図5に示した初期状態のEEPROM108に対し、フォーマット処理を実行する。

【0059】このフォーマット実行処理では、まず、命令を受けた数の空きセクタをファイル領域に確保し、その確保した空きセクタに対応する管理セクタFをFAT領域に生成する。その後、システムディレクトリ（SYS）とMFディレクトリ（MF）とを確保した空きセクタに生成する。以下、確保する空きセクタ数として256セクタが指示された場合について説明する。

【0060】（管理セクタのリンク付け）まず、ファイル領域において、論理アドレス「#0000」～「#00FF」の

256セクタ分の空きセクタを確保すべく、確保する空きセクタに各々対応する管理セクタをFAT領域に生成する。具体的には、FAT領域において「#0000」の管理セクタに、次の論理アドレスへのリンク情報として「#0001」を登録し、続く「#0001」には次の論理アドレスのリンク情報として「#0002」を登録する。同様にして、各管理セクタに次の論理アドレスへのリンク情報を登録し、最終セクタに対応する論理アドレス「#00FF」の管理セクタには、最終コードを示す「FFFF」を登録する。このようにして、各管理セクタに次の論理アドレスへのリンク情報を登録することにより、論理アドレス「#0000」～「#00FF」の連続する256セクタを空きセクタとして確保する。

【0061】（システムディレクトリ（SYS）の生成）続いて、CPU105は、図4に示したファイル構造において最上位に位置するシステムディレクトリ（SYS）の生成処理を実行する（図4のSP1）。まず、ファイル領域の先頭物理アドレス「H'000000」～「H'00000F」からなる1セクタ目に、システムディレクトリを生成する。これは、1セクタ目のディレクトリ生成・削除フラグ格納エリア（以下、単にFLGと記載する）に、ディレクトリを作成したことを示すための「01」を書き込み、続いて、先頭セクタ番号（先頭論理アドレス）格納エリアTOP（以下、単にTOPと記載する）に、現在の空きセクタの先頭論理アドレスである「#0000」を書き込み、最終セクタ番号格納エリアBTM（以下、単にBTMと記載する）に空きセクタの最終論理アドレスである「#00FF」を書き込み、ファイルサイズ格納エリアVLM（以下、単にVLMと記載する）に現在の空きセクタ数である「0100」を書き込む。なお、このセクタ数も16進表記である。これにより、図6に示すように、システムディレクトリが1セクタ目に生成され、現在ファイル領域における「#0000」～「#00FF」までの「256」セクタが空き領域として確保された旨が示される。

【0062】（MFディレクトリ（MF）の生成）続いて、CPU105はMFディレクトリの生成処理を行う（図4のSP2）。まず、現在確保されている空きセクタの内、先頭の1セクタをMFディレクトリ生成用に確保するため、システムディレクトリのTOPに記載されている論理アドレスを読み取る（図6参照）。この結果、論理アドレス「#0000」を取得すると、この論理アドレスで識別されるFAT領域の管理セクタに、最終コードである「FFFF」を登録し（図7参照）、当該論理アドレスで識別されるセクタを上記した空きセクタのリンクから切り離す。これにより、論理アドレス「#0000」のセクタをMFディレクトリ生成用に確保することができる。

【0063】続いて、256あった空きセクタの内、1セクタをMFディレクトリ生成用に確保した旨をシステムディレクトリに反映させる。即ち、MFディレクトリ

生成用に1セクタ確保することにより、空きセクタの先頭論理アドレスは「#0000」から「#0001」へ移行し、セクタ数は1つ減少して「255」となる。従って、この旨をシステムディレクトリに登録するべく、システムディレクトリのTOPに「#0001」を、VLMにセクタ数「255」を示す「#00FF」を書き込む。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0064】続いて、CPUは、確保した論理アドレス「#0001」のセクタにMFディレクトリを生成する。即ち、ディレクトリが生成された旨を示すためにFLGに「01」を登録する。なお、MFディレクトリ配下には、まだファイルが存在しないため、MFディレクトリのTOP、BTM、VLMには初期値のままである。上述した処理が行われることにより、EEPROM108のメモリマップは図7に示すような状態となる。

【0065】(リセット応答EFディレクトリ(R-EF)の生成) 続いて、CPUはMFディレクトリ(MF)の配下にリセット応答EFディレクトリ(R-EF)を生成する処理を行う(図4のSP3)。まず、空きセクタの内、先頭の1セクタをリセット応答EFディレクトリ生成用に確保するため、システムディレクトリのTOPに記載されている論理アドレス「#0001」を読み取り、この論理アドレス「#0001」の管理セクタに、最終コードである「FFFF」を登録する。これにより、論理アドレス「#0001」で識別される1セクタを空きセクタのリンケージから切り離し、リセット応答EFディレクトリ生成用に確保することができる。

【0066】続いて、論理アドレス「#0001」のセクタを確保したことにより、空きセクタの先頭論理アドレスが「#0002」に移行し、空きセクタ数も1減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリのTOPに「#0002」を、VLMにセクタ数「254」を示す「#00FE」を登録する。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0067】続いて、CPUはリセット応答EFディレクトリをMFディレクトリ配下に生成する旨をMFディレクトリに反映させる処理を行う。具体的には、まず、MFディレクトリの配下に既にファイルが存在するかどうかを判断するために、MFディレクトリのBTMに「FF」が登録されているかどうかを判断する。この結果、図7から分かるように、MFディレクトリのBTMには「FF」が登録されているため、CPU105は、リセット応答EFディレクトリ用に確保した1セクタの情報をMFディレクトリにそのまま反映させる。

【0068】具体的には、MFディレクトリのTOPに「#0001」を、BTMに「#0001」を、VLMに「0001」を書き込む。これにより、MFディレクトリ配下にファイルが生成され、且つ、そのファイルは論理アドレス「#0001」

のセクタに格納されていることを示すことができる。

【0069】続いて、CPUは、確保した論理アドレス「#0001」のセクタにリセット応答EFディレクトリを生成するべく、論理アドレス「#0001」のセクタのFLGに「01」を登録する。なお、リセット応答EFディレクトリ配下には、まだファイル等が存在しないため、リセット応答EFディレクトリのTOP、BTM、VLMは初期値のままである。上述した処理が行われることにより、EEPROM108のメモリマップは図8に示すような状態となる。

【0070】(EF1の書き込み) 続いて、CPUはリセット応答EFディレクトリ(R-EF)の下にEF1を書き込む処理を行う(図4のSP4)。まず、システムディレクトリのTOPに格納されているアドレス「#0002」を読み取り、この論理アドレスを先頭論理アドレスとして、これから書き込むEF1のサイズに応じたセクタ数を確保する。ここでは、2セクタ、即ち「#0002」～「#0003」をEF1格納用に確保するべく、確保した2セクタの最終論理アドレス「#0003」に対応するFAT領域の管理セクタに、最終コードを示す「FFFF」を登録する。これにより、論理アドレス「#0002」～「#0003」で識別される2セクタを空きセクタのリンケージから切り離し、EF1格納用に確保することができる。

【0071】続いて、論理アドレス「#0002」～「#0003」の2セクタを確保したことにより、空きセクタの先頭論理アドレスが「#0004」に移行し、空きセクタ数も2減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリのTOPに「#0004」を、VLMにセクタ数「252」を示す「#00FC」を登録する。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0072】続いて、CPUはEF1をリセット応答EF配下に生成する旨をリセット応答EFディレクトリに反映させる処理を行う。具体的には、まず、リセット応答EFディレクトリのBTMに「FFFF」が登録されているかどうかを判断する。この結果、図8から分かるように、BTMには「FFFF」が登録されているため、CPUは、EF1格納用に確保した2セクタの情報をリセット応答EFディレクトリにそのまま反映させる。

【0073】具体的には、リセット応答EFディレクトリのTOPに「#0002」、BTMに「#0003」、VLMに「0002」を書き込む。これにより、リセット応答EF配下にファイルが生成され、且つ、そのファイルは論理アドレス「#0002」～「#0003」のセクタに格納されていることを示すことができる。続いて、CPUは、確保した論理アドレス「#0002」～「#0003」にEF1を書き込む。上述した処理が行われることにより、EEPROM108のメモリマップは図9に示すような状態となる。

【0074】(DF名管理EFディレクトリ(EF-EF)の生成) 続いて、CPU105はMFディレクトリ(MF)

F) の配下に DF 名管理 EF ディレクトリ (DF-EF) を生成する (図 4 の SP5)。まず、現在確保されている空きセクタの内、先頭の 1 セクタを DF 管理 EF ディレクトリ生成用に確保するため、システムディレクトリの TOP に記載されている論理アドレスを読み取る (図 9 参照)。この結果、論理アドレス「#0004」を取得すると、この論理アドレスで識別される FAT 領域の管理セクタに、最終コードである「FFFF」を登録し、当該論理アドレスで識別されるセクタを上述した空きセクタのリンクから切り離す。これにより、論理アドレス「#0004」のセクタを DF 名 EF ディレクトリ生成用に確保することができる。

【0075】続いて、論理アドレス「#0004」のセクタを確保したことにより、空きセクタの先頭論理アドレスが「#0005」に移行し、空きセクタ数も 1 減少したの、その旨をシステムディレクトリに反映させるべく、システムディレクトリの TOP に「#0005」を、VLM にセクタ数「251」を示す「00FB」を登録する (図 10 参照)。なお、最終セクタについては、変更がないので、BTM は「#00FF」のままである。

【0076】続いて、CPU105 は DF 名管理 EF ディレクトリを MF ディレクトリ配下に生成する旨を MF ディレクトリに反映させる処理を行う。具体的には、まず、MF ディレクトリの BTM に「FFFF」が登録されているか否かを判断する。この結果、図 9 から分かるように、MF ディレクトリの BTM には「FFFF」以外の情報が登録されているため、CPU は、MF ディレクトリの BTM に書き込まれている「#0001」の情報を、今回確保したセクタの最終論理アドレスである「#0004」に更新し、また、VLM の情報を「0002」に更新することにより、新たに確保したセクタ 1 を反映させる (図 10 参照)。

【0077】続いて、CPU105 は、MF ディレクトリの BTM の前回値 (更新する前の値) である論理アドレス「#0001」で識別される FAT 領域の管理セクタに、今回、DF 名管理 EF ディレクトリ生成用に確保したセクタの先頭論理アドレスである「#0004」を書き込む。即ち、FAT 領域の論理アドレス「#0001」の管理セクタの情報を、「FFFF」から「#0004」に書き換える。

【0078】これにより、図 10 に示すように、CPU105 は、MF ディレクトリの TOP に書き込まれている情報「#0001」の情報に基づいて、FAT 領域を検索することにより、論理アドレス「#0001」→「#0004」と追うことができ、論理アドレス「#0001」、「#0004」の情報 MF 配下に存在するということを認識することができる。

【0079】続いて、CPU105 は、確保した論理アドレス「#0004」のセクタに DF 名管理 EF ディレクトリを生成するべく、論理アドレス「#0004」のセクタの FLG に「01」を登録する。なお、DF 名管理 EF ディレクトリ配下には、まだファイル等が存在しないため、DF

名管理 EF ディレクトリの TOP、BTM、VLM は初期値のままである。上述した処理が行われることにより、EEPROM108 のメモリマップは図 10 に示すような状態となる。

【0080】(DF1 ディレクトリ (DF1) の生成) 続いて、CPU105 は MF ディレクトリ (MF) の配下に DF1 ディレクトリ (DF1) を生成する (図 4 の SP6)。まず、現在確保されている空きセクタの内、先頭の 1 セクタを DF1 ディレクトリ生成用に確保するため、システムディレクトリの TOP に記載されている論理アドレスを読み取る (図 10 参照)。この結果、論理アドレス「#0005」を取得すると、この論理アドレスで識別される FAT 領域の管理セクタに、最終コードである「FFFF」を登録し、当該論理アドレスで識別されるセクタを上述した空きセクタのリンクから切り離す。これにより、論理アドレス「#0005」のセクタを DF1 ディレクトリ生成用に確保することができる。

【0081】続いて、論理アドレス「#0005」のセクタを確保したことにより、空きセクタの先頭論理アドレスが「#0006」に移行し、空きセクタ数も 1 減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリの TOP に「#0006」を、VLM にセクタ数「250」を示す「00FA」を登録する。なお、最終セクタについては、変更がないので、BTM は「#00FF」のままである。

【0082】続いて、CPU105 は DF1 ディレクトリを MF ディレクトリ配下に生成する旨を MF ディレクトリに反映させる処理を行う。具体的には、まず、MF ディレクトリの BTM に「FFFF」が登録されているか否かを判断する。この結果、図 10 から分かるように、MF ディレクトリの BTM には「FFFF」以外の情報が登録されているため、CPU は、MF ディレクトリの BTM に書き込まれている「#0004」の情報を、今回確保したセクタの最終論理アドレスである「#0005」に更新し、また、VLM の情報を「0003」に更新することにより、新たに確保したセクタ 1 を反映させる (図 11 参照)。

【0083】続いて、CPU は、MF ディレクトリの BTM の前回値 (更新する前の値) である論理アドレス「#0004」で識別される FAT 領域の管理セクタに、今回、DF1 ディレクトリ生成用に確保したセクタの先頭論理アドレスである「#0005」を書き込む。即ち、FAT 領域の論理アドレス「#0004」の情報を、「FFFF」から「#0005」に書き換える。

【0084】これにより、図 11 に示すように、CPU は、MF ディレクトリの TOP に書き込まれている情報「#0001」の情報に基づいて、FAT 領域を検索することにより、論理アドレス「#0001」→「#0004」→「#0005」と追うことができ、論理アドレス「#0001」、「#0004」、「#0005」で識別されるセクタに格納されているファイル等が MF 配下に存在するということを認識するこ

とができる。

【0085】続いて、CPUは、確保した論理アドレス「#0005」のセクタにDF1ディレクトリを生成するべく、論理アドレス「#0005」のセクタのFLGに「01」を登録する。なお、DF1ディレクトリ配下には、まだファイル等が存在しないため、このディレクトリのTOP、BTM、VLMは初期値のままである。上述した処理が行われることにより、EEPROM108のメモリマップは図11に示すような状態となる。

【0086】(DF1のファイル名登録) 続いて、DF1ディレクトリ (DF) のファイル名をDF名管理EF (DF-EF) の配下へ書き込む処理を行う (図4のSP7)。DFには、ファイル名を付与することができる。ここでは、DFファイルに付与されたファイル名を専用に管理するファイルであるDF名管理EF配下に、DF1ディレクトリに付与されたDF名を登録する処理を行う。

【0087】まず、システムディレクトリTOPに格納されているアドレス「#0006」を読み取り、この論理アドレスを先頭論理アドレスとして、2セクタをDF1のファイル名登録用に確保するため、この論理アドレス「#0006」から2セクタ目に当たる論理アドレス「#0007」の管理セクタに、最終コードである「FFFF」を登録する。これにより、論理アドレス「#0006」～「#0007」で識別される2セクタを空きセクタのリンケージから切り離し、DF1のファイル名登録用に確保することができる。

【0088】続いて、論理アドレス「#0006」～「#0007」の2セクタを確保したことにより、空きセクタの先頭論理アドレスが「#0008」に移行し、空きセクタ数も2減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリのTOPに「#0008」を、VLMにセクタ数「248」を示す「00F8」を登録する。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0089】続いて、CPU105はDF1のファイル名をDF名管理EF配下に登録する旨をDF名管理EFディレクトリに反映させる処理を行う。具体的には、まず、DF名管理EFディレクトリのBTMに「FFFF」が登録されているか否かを判断する。この結果、図11から分かるように、DF名管理EFディレクトリのBTMには「FFFF」が登録されているため、CPUは、DF1のファイル名登録用に確保した2セクタの情報をDF名管理EFディレクトリにそのまま反映させる。具体的には、DF名管理EFディレクトリのTOPに「#0006」を、BTMに「#0007」を、VLMに「0002」を書き込む。これにより、DF名管理EF配下にDF1のファイル名が登録され、且つ、そのファイルは論理アドレス「#0006」～「#0007」のセクタに格納されていることを示すことができる。

【0090】続いて、CPUは、確保した論理アドレス「#0006」～「#0007」のセクタのFLGに「01」を登録するとともに、DFディレクトリの先頭アドレス格納領域TOP_DにDF1の先頭論理アドレス「#0005」を、またファイル名格納領域にDF1のファイル名を書き込む。なお、このファイル名登録用のセクタのレイアウトについては、図3に示した通りである。このような処理を行うことにより、DF1の先頭論理アドレス及びファイル名がDF名管理EFに登録され、以後、管理される。これにより、ファイル名によるDF検索を可能とし、DF1の先頭論理アドレス情報を得ることができ、DF1にアクセスすることができる。上述した処理が行われることにより、EEPROM108のメモリマップは図12に示すような状態となる。

【0091】(EF2ディレクトリ (EF2) の生成) 続いて、CPUはDF1ディレクトリ (DF1) の配下にEF2ディレクトリを生成する処理を行う (図4のSP8)。まず、空きセクタの内、先頭の1セクタをEF2ディレクトリ (EF2) 生成用に確保するため、システムディレクトリのTOPに記載されている論理アドレス「#0008」を読み取り、この論理アドレス「#0008」の管理セクタに、最終コードである「FFFF」を登録する。これにより、論理アドレス「#0008」で識別される1セクタを空きセクタのリンケージから切り離し、EF2ディレクトリ生成用に確保することができる。

【0092】続いて、論理アドレス「#0008」のセクタを確保したことにより、空きセクタの先頭論理アドレスが「#0009」に移行し、空きセクタ数も1減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリのTOPに「#0009」を、VLMにセクタ数「247」を示す「00F7」を登録する。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0093】続いて、CPU105はEF2ディレクトリ (EF2) をDF1ディレクトリ (DF1) 配下に生成する旨をDF1ディレクトリに反映させる処理を行う。具体的には、まず、DF1ディレクトリのBTMに「FFFF」が登録されているか否かを判断する。この結果、図12から分かるように、DF1ディレクトリのBTMには「FFFF」が登録されているため、CPUは、EF2ディレクトリ生成用に確保した1セクタの情報をDF1ディレクトリにそのまま反映させる。具体的には、DF1ディレクトリのTOPに「#0008」を、BTMに「#0009」を、VLMに「0001」を書き込む (図13参照)。これにより、DF1配下に下位ディレクトリが生成され、且つ、その下位ディレクトリは論理アドレス「#0008」のセクタに格納されていることを示すことができる。

【0094】続いて、CPU105は、確保した論理アドレス「#0008」のセクタにEF2ディレクトリを生成するべく、論理アドレス「#0008」のセクタのFLGに「0

1]」を登録する。なお、EF 2ディレクトリ配下には、まだファイル等が存在しないため、EF 2ディレクトリのTOP、BTM、VLMは初期値のままである。上述した処理が行われることにより、EEPROM108のメモリマップは図13に示すような状態となる。

【0095】(EF 2の書き込み) 続いて、CPUはEF 2ディレクトリ (EF2) の下にEF 2を書き込む処理を行う (図4のSP 9)。まず、システムディレクトリのTOPに格納されているアドレス「#0009」を読み取り、この論理アドレスを先頭論理アドレスとして、これから書き込むEF 2のサイズに応じたセクタ数を確保する。ここでは、2セクタ、即ち「#0009」～「#000A」をEF 2格納用に確保するべく、確保した最終論理アドレス「#000A」に対応するFAT領域の管理セクタに最終コードを示す「FFFF」を登録する。これにより、論理アドレス「#0009」～「#000A」で識別される2セクタを空きセクタのリンケージから切り離し、EF 2格納用に確保することができる。

【0096】続いて、論理アドレス「#0009」～「#000A」の2セクタを確保したことにより、空きセクタの先頭論理アドレスが「#000B」に移行し、空きセクタ数も2減少したので、その旨をシステムディレクトリに反映させるべく、システムディレクトリのTOPに「#000B」を、VLMにセクタ数「245」を示す「00F5」を登録する。なお、最終セクタについては、変更がないので、BTMは「#00FF」のままである。

【0097】続いて、CPU105はEF 2をEF 2ディレクトリ配下に生成する旨をEF 2ディレクトリに反映させる処理を行う。具体的には、まず、EF 2ディレクトリのBTMに「FFFF」が登録されているかを判断する。この結果、図13から分かるように、BTMには「FFFF」が登録されているため、CPUは、EF 2格納用に確保した2セクタの情報をEF 2ディレクトリにそのまま反映させる。

【0098】具体的には、EF 2ディレクトリのTOPに「#0009」、BTMに「#000A」、VLMに「0002」を書き込む (図4参照)。これにより、EF 2ディレクトリ配下にファイルが生成され、且つ、そのファイルは論理アドレス「#0009」～「#000A」のセクタに格納されていることを示すことができる。続いて、CPU105は、確保した論理アドレス「#0009」～「#000A」にEF 2を書き込む。上述した処理が行われることにより、EEPROM108のメモリマップは図14に示すような状態となる。

【0099】続いて、CPUは同様の方法で、DF 1ディレクトリ (DF1) の配下にDF 2ディレクトリ (DF2) を生成する処理 (図4のSP 10)、DF 2のファイル名をDF名管理EFディレクトリ (DF-DF) の配下へ書き込む処理 (図4のSP 11)、DF 2ディレクトリ (DF2) の配下にEF 3ディレクトリ (EF3) を生成する

処理 (図4のSP 12)、EF 3ディレクトリ (EF3) の下にEF 3を書き込む処理 (図4のSP 13)、DF 2ディレクトリ (DF2) の配下にEF 4ディレクトリ (EF4) を生成する処理 (図4のSP 14)、EF 4ディレクトリ (EF4) の下にEF 4を書き込む処理 (図4のSP 15) を行う。この結果、EEPROM108のメモリマップは図15に示すような状態となる。

【0100】(5) ATR情報について
一般的にICカードは機種によりその伝送プロトコルが異なるので、ICカードは伝送プロトコル等を外部装置であるカードR/Wに知らせるためリセット信号によるリセット解除を認識後、ATR (Answer To Reset) と称する初期応答データをカードR/Wに出力する。その後、ICカードは命令待ち状態になる。この場合、初期応答データは、ICカードの伝送プロトコル使用情報とICカード固有の情報とから構成されており、ICカードの製造時に固有パラメータとしてROMに予め登録されている。

【0101】ところで、ICカードを利用者に対して発行した後に、外部装置等のアプリケーションが変更された場合には、ICカード内に登録されたATR情報を更新する必要性が生じる。この場合、ROM内に書き込まれたATR情報を更新することは不可能であるため、情報の書き換えが可能であるEEPROM内の特定のファイル内に変更後のATR情報を格納し、以降、外部装置からリセット信号を受け付けたときには、EEPROM内の特定のファイル内に登録されたATR情報を出力する。

【0102】ここで、本発明の一実施形態に係るEEPROMのメモリ階層構造を図16に示す。ここで、リセット応答EFは、ATR情報書き込み専用ファイルとしてEEPROM内に設けられたファイルである。また、認証EFには、リセット応答EF内のデータを書き換える際に必要となる認証キーの情報等が格納されている。なお、以下説明する本実施形態に係るATR情報書き換え処理は、上述した接触型ICカード、非接触型ICカード、複合型ICカードのいずれにも適用することができる。なお、以下の説明におけるEEPROMとは、各種ICカードのICチップ内に実装されているEEPROMのことであり、例えば、接触型ICカードに本発明を適用する場合には、図21に示したEEPROM108のことであり、非接触型ICカードに本発明を適用する場合には、EEPROM114のことである。同様に、複合型ICカードでは、EEPROM123のことである。

【0103】以下、上述したようなファイル構造を持つEEPROMのリセット応答EFに対して、新規にATR情報を書き込む処理について図17を参照して説明する。なお、前提としてEEPROMのリセット応答EFは、初期状態であるものとする。

【0104】まず、外部装置はICカードを起動させるためにリセット信号を出力する(図17のステップSP101)。これにより、リセット信号を受け取ったICカード内のCPUは、ROM又はEEPROM内に格納されているATR情報の読み出し処理を行い、読み出したATR情報を外部装置へ送信する(ステップSP102)。

【0105】ここで、上記ATR情報の読み出し処理について図18を参照して説明する。まず、CPUは、EEPROM内のリセット応答EFに付加されているID名に基づいて、リセット応答EFを検索する(図18のステップSP201)。続いて、リセット応答EFのデータ部が初期状態であるか否かを判断する。(ステップSP202)。この結果、リセット応答EFのデータ部は初期状態(例えば、データが全て「F」の状態)であるので、CPUは、ROM内に登録されているATR情報を読み出し、読み出したATR情報を出力する(ステップSP203、図17のステップSP102)。

【0106】続いて、外部装置はATR情報を受け取ると、続いて、リセット応答EFを選択する指示をICカードに対して出力する(図17のステップSP103)。この指示を受けて、ICカード内のCPUはリセット応答EFを選択し、その旨を外部装置へ通知する(ステップSP104)。続いて、外部装置は、ICカードに対して新たなATR情報①の書き込み要求を出力する。この要求を受けたICカード内のCPUは、該要求と共に受け取ったATR情報①をステップSP104において選択したリセット応答EFディレクトリの配下に書き込む処理を実行する(上述の「Fデータ1等の書き込み」参照)。これにより、リセット応答EFディレクトリ配下に新たなATR情報①を書き込むことが可能となる。

【0107】次に、上述したような処理が行われることにより、EEPROM内のリセット応答EFに書き込まれたATR情報①を更に新しいATR情報②に書き換える処理について図19を参照して説明する。まず、外部装置はICカードを起動させるためにリセット信号を出力する(図19のステップSP111)。これにより、リセット信号を受け取ったICカード内のCPUは、図17に示した処理を行うことにより、ROM内及びEEPROMのリセット応答EF内に格納されているATR情報①を読み出し、このATR情報①を外部装置へ出力する(ステップSP112)。

【0108】続いて、外部装置はリセット応答EFを選択する旨の指示を出力する(ステップSP113)。これにより、ICカード内のCPUはリセット応答EFを選択し、その旨を外部装置へ通知する(ステップSP114)。続いて、外部装置はリセット応答EFのデータ部に格納されているATR情報①の読み出し要求を出力する(ステップSP115)。これにより、ICカード

のCPUはリセット応答EFのデータ部からATR情報①を読み出し、出力する(ステップSP116)。続いて、外部装置はATR情報①の書き込み要求を出力する(ステップSP117)。書き込み要求を受け取ったICカードのCPUは、鍵の認証処理を行った後でなければ、書き込みを許可できないため、書き込み要求を拒否する旨の情報、即ち、書き込み拒否のエラー通知を外部装置へ出力する(ステップSP118)。

【0109】続いて、外部装置はATR情報②の書き換え要求を出力する(ステップSP119)。書き込み要求を受け取ったICカードのCPUは、鍵の認証処理を行った後でなければ、書き換えを許可することができないため、書き換え要求を拒否する旨の通知、即ち書き換え拒否のエラー通知を出力する(ステップSP120)。続いて、外部装置は、自己が保有している認証鍵(KEY1)を発行(出力)する(ステップSP121)。この認証鍵の情報を受け取ったICカードのCPUは、自己が有している認証キーの情報と照合を行い、両者が一致、又は正当な組み合わせであることを確認すると、書き換えアクセス許可をRAMに記憶し、書き換え要求を許可する旨を外部装置へ出力する(ステップSP122)。

【0110】続いて、書き換え要求を許可する旨の通知を受けた外部装置は、ATR情報②の書き換え要求を出力する(ステップSP123)。これにより、ICカードのCPUは、リセット応答EFのデータ部に格納されているATR情報①をATR情報②に書き換える(ステップSP124)。上述したような通信が外部装置とICカードとの間で行われることにより、リセット応答EF内のATR情報を書き換えることが可能となる。

【0111】以上、この発明の実施形態を図面を参照して詳述してきたが、具体的な構成はこの実施形態に限られるものではなく、この発明の要旨を逸脱しない範囲の設計等も含まれる。

【0112】

【発明の効果】以上説明したように、本発明のICカードによれば、不揮発性メモリ内に、新たな初期応答データを書き込むための初期応答データ書き込み専用ファイルを設けるので、ICカード発行後において、ROM内に予め登録されている初期応答データを変更する必要がある場合には、書き換え可能な不揮発性メモリに新たな初期応答データを書き込むことにより、初期応答データの更新を行うことができる。これにより、外部機器に対して変更後の即ち、最新のICカードの初期情報を通知することが可能となる。また、初期応答データを専用格納する初期応答データ書き込み専用ファイルを設けることにより、新たな初期応答データを容易に読み出すことができ、迅速な外部装置とのデータ通信を実現することができる。

【0113】また、請求項2に記載のICカードによれ

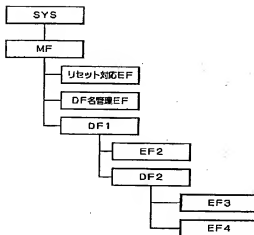
ば、外部からリセット信号を受信した場合に、初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれているかを判断し、初期応答データ書き込み専用ファイル内に新たな初期応答データが書き込まれている場合には、ROM内に書き込まれている初期応答データと初期応答データ書き込み専用ファイル内の新たな初期応答データとを読み出して、外部に出力する。このようにROM及び不揮発性メモリの双方から初期応答データを読み出すので、初期応答データ書き込み専用ファイルには、元の初期応答データに対して変更が生じた部分だけを登録しておけばよい。これにより、初期応答データ用ファイルに登録するデータ量を軽減させることができ、初期応答データの読み出し処理を迅速に行うことが可能となる。

【図面の簡単な説明】

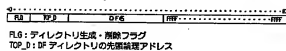
- 【図1】 本発明の一実施形態に係るICカードのファイル階層構造を示す図である。
 【図2】 ディレクトリを生成するセクタのレイアウトの一例を示す図である。
 【図3】 DFのファイル名を格納するセクタのレイアウトの一例を示す図である。
 【図4】 図1に示したファイル階層構造をより具体的に示した図である。
 【図5】 初期状態におけるEEPROMのメモリマップを示す図である。
 【図6】 システムディレクトリを生成したときのEEPROMのメモリマップを示す図である。
 【図7】 MFを生成したときのEEPROMのメモリマップを示す図である。
 【図8】 リセット応答EFディレクトリを生成したときのEEPROMのメモリマップを示す図である。
 【図9】 EF1を格納したときのEEPROMのメモリマップを示す図である。
 【図10】 DF名管理EFディレクトリを生成したときのEEPROMのメモリマップを示す図である。
 【図11】 DF1ディレクトリを生成したときのEEPROMのメモリマップを示す図である。
 【図12】 DF1のファイル名を格納したときのEEPROMのメモリマップを示す図である。
 【図13】 EF2ディレクトリを生成したときのEEPROMのメモリマップを示す図である。
 【図14】 EF2を格納したときのEEPROMのメモリマップを示す図である。
 【図15】 図1に示すファイル階層構造をEEPROMに作成したときのメモリマップを示す図である。
 【図16】 本発明の一実施形態に係るICカードのEEPROMのメモリ階層構造を示す図である。

- * 【図17】 EEPROMのリセット応答EFに対して、新規にATR情報を書き込む処理について示したフローチャートである。
 【図18】 ATR情報の読み出し処理について示したフローチャートである。
 【図19】 ATR情報①を更に新しいATR情報②に書き換える処理について示したフローチャートである。
 【図20】 代表的な非接触型ICカードの概観図である。
 【図21】 接触型ICカードの電気構成図である。
 【図22】 代表的な非接触型ICカードの概観図である。
 【図23】 非接触型ICカードの電気回路構成を示す図である。
 【図24】 複合型ICカードの電気回路構成を示す図である。
 【図25】 複合型ICカードの構造を示す図である。
 【図26】 複合型ICカードの平面図である。
 【図27】 複合型ICカードにおけるコイルの複合型ICカード内部に於ける実装位置を示す図である。
 【図28】 外部装置が放射する高周波電磁界の周波数が、図39及び図40の場合に比べて1桁乃至2桁低い場合に選んだコイルの巻き方を示した図である。
 【図29】 代表的なICカードシステムの構成を示す図である。
 【図30】 接触型のインターフェースが使用される場合のICカードとカードR/Wとの通信を説明するための図である。
 【図31】 非接触型のインターフェースが使用される場合のICカードとカードR/Wとの通信を説明するための図である。
 【図32】 ICカードの製造から発行までの過程を示す図である。
 【図33】 原国際規格ISO/IEC7816シリーズにより規定されている基本的なファイル階層構造を示す図である。
 【符号の説明】
 Y セクタ (区画)
 F 管理セクタ (管理区画)
 105, 111, 121 CPU (中央処理装置)
 106, 112, 122 ROM
 107, 113, 124 RAM
 108, 114, 123 EEPROM
 101, 125 端子電極
 116, 126 共振回路部
 115, 127 RF回路
 102, 110 ICチップ

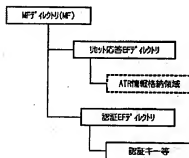
【図 1】



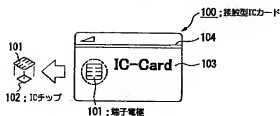
【図 3】



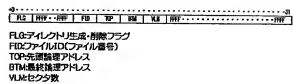
【図 16】



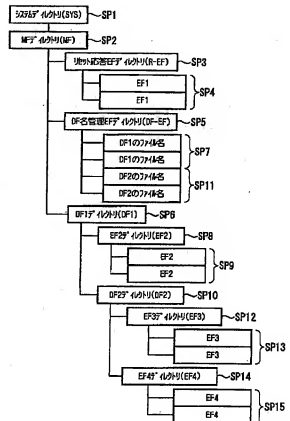
【図 20】



【図 2】



【図 4】



【図 5】

Y:セクタ
(32バイト)

〈ファイル領域〉

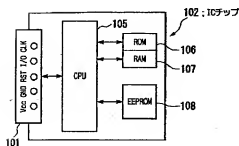
物理アドレス	論理アドレス	FLS	TP	BTM	VM
H 00000	—	FF	FFF	FFF	FFF
H 00000	#0000	FF	FFF	FFF	FFF
H 00040	#0001	FF	FFF	FFF	FFF
H 00080	#0002	FF	FFF	FFF	FFF
H 00080	#0003	FF	FFF	FFF	FFF
H 00040	#0004	FF	FFF	FFF	FFF
H 00000	#0005	FF	FFF	FFF	FFF
H 00000	#0006	FF	FFF	FFF	FFF
H 00000	#0007	FF	FFF	FFF	FFF
H 000100	#0008	FF	FFF	FFF	FFF
H 000120	#0009	FF	FFF	FFF	FFF
H 000140	#000A	FF	FFF	FFF	FFF
H 000160	#000B	FF	FFF	FFF	FFF
H 000180	#000C	FF	FFF	FFF	FFF
H 0001A0	#000D	FF	FFF	FFF	FFF
H 0001C0	#000E	FF	FFF	FFF	FFF
H 0001E0	#000F	FF	FFF	FFF	FFF
H 000200	#0010	FF	FFF	FFF	FFF
H 000220	#0011	FF	FFF	FFF	FFF
H 000240	#0012	FF	FFF	FFF	FFF
H 000260	#0013	FF	FFF	FFF	FFF
H 000280	#0014	FF	FFF	FFF	FFF
H 0002A0	#0015	FF	FFF	FFF	FFF
H 0002C0	#0016	FF	FFF	FFF	FFF
H 0002E0	#0017	FF	FFF	FFF	FFF
H 000300	...	FFF

F F F; 管理セクタ(2バイト)

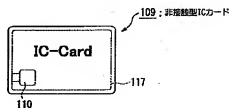
〈FAT領域〉

H 05000	FFF	FFF	FFF	FFF	FFF	FFF	FFF
H 05010	FFF	FFF	FFF	FFF	FFF	FFF	FFF
H 05020	FFF	FFF	FFF	FFF	FFF	FFF	FFF
:
:
:
:
:

【図 2 1】



【図 2 2】



【圖6】

SYS	物部 12	部部 12	FLG	TOP	BTM	WM
MF>	H 00000	-	01 XX--XXXX	H0000	H00FF	0100 XXXX 1/2561271情報 XXXX
	H 00000	H0000	FF	FFFF	FFFF	FFFF
	H 00040	H0001	FF	FFFF	FFFF	FFFF
	H 00080	H0002	FF	FFFF	FFFF	FFFF
	H 00080	H0003	FF	FFFF	FFFF	FFFF
	H 00040	H0004	FF	FFFF	FFFF	FFFF
	H 00000	H0005	FF	FFFF	FFFF	FFFF
	H 00020	H0006	FF	FFFF	FFFF	FFFF
	H 00010	H0007	FF	FFFF	FFFF	FFFF
	H 00010	H0008	FF	FFFF	FFFF	FFFF
	H 00040	H0009	FF	FFFF	FFFF	FFFF
	H 00010	H000A	FF	FFFF	FFFF	FFFF
	H 00080	H000B	FF	FFFF	FFFF	FFFF
	H 00040	H000C	FF	FFFF	FFFF	FFFF
	H 00010	H000D	FF	FFFF	FFFF	FFFF
	H 00010	H000E	FF	FFFF	FFFF	FFFF
	H 00020	H000F	FF	FFFF	FFFF	FFFF
	H 00020	H0010	FF	FFFF	FFFF	FFFF
	H 00040	H0011	FF	FFFF	FFFF	FFFF
	H 00020	H0012	FF	FFFF	FFFF	FFFF
	H 00080	H0013	FF	FFFF	FFFF	FFFF
	H 00040	H0014	FF	FFFF	FFFF	FFFF
	H 00020	H0015	FF	FFFF	FFFF	FFFF
	H 00020	H0016	FF	FFFF	FFFF	FFFF
	H 00030	H0017	FF	FFFF	FFFF	FFFF
:	:	:	FFF	FFF
:	:	:	FFF	FFF
:	:	:	FFF	FFF

[illegible]

【図9】

	物理アドレス	論理アドレス	FLG	TOP	BIN	VLM	
SYS	H 000000	—	01	XX -- -XXX	#0004	#00FF	00FC XXXX 1分1分1分1分 XXXX
MF	H 000020	#0000	01	XX -- -XXX	#0001	#0001	XXXX MF 1分1分1分1分 XXXX
R-EF	H 000040	#0001	01	XX -- -XXX	#0002	#0003	0002 XXXX EF 1分1分1分1分 XXXX
	H 000060	#0002					
	H 000080	#0003					
	H 0000A0	#0004	FF		FFFF	FFFF	FFFF
	H 0000C0	#0005	FF		FFFF	FFFF	FFFF
	H 0000E0	#0006	FF		FFFF	FFFF	FFFF
	H 000100	#0007	FF		FFFF	FFFF	FFFF
	H 000120	#0008	FF		FFFF	FFFF	FFFF
	H 000140	#0009	FF		FFFF	FFFF	FFFF
	H 000160	#000A	FF		FFFF	FFFF	FFFF
	H 000180	#000B	FF		FFFF	FFFF	FFFF
	H 0001A0	#000C	FF		FFFF	FFFF	FFFF
	H 0001C0	#000D	FF		FFFF	FFFF	FFFF
	H 0001E0	#000E	FF		FFFF	FFFF	FFFF
	H 000200	#000F	FF		FFFF	FFFF	FFFF
	H 000220	#0010	FF		FFFF	FFFF	FFFF
	H 000240	#0011	FF		FFFF	FFFF	FFFF
	H 000260	#0012	FF		FFFF	FFFF	FFFF
	H 000280	#0013	FF		FFFF	FFFF	FFFF
	H 0002A0	#0014	FF		FFFF	FFFF	FFFF
	H 0002C0	#0015	FF		FFFF	FFFF	FFFF
	H 0002E0	#0016	FF		FFFF	FFFF	FFFF
	H 000300	#0017	FF		FFFF	FFFF	FFFF
	:	:	FFFF	FFFF	FFFF	FFFF
	:	:	FFFF	FFFF	FFFF	FFFF
	:	:	FFFF	FFFF	FFFF	FFFF

H 050000

H 005010

H 005020

:

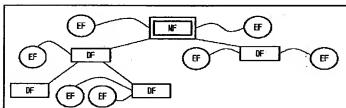
:

:

:

#0000	#0001	#0002	#0003	#0004	#0005	#0006	#0007
FFFF	FFFF	#0003	FFFF	#0005	#0006	#0007	#0008
#0009	#000A	#000B	#000C	#000D	#000E	#000F	#0010
#0011	#0012	#0013	#0014	#0015	#0016	#0017	#0018
#0019	#001A	#001B	#001C	#001D	#001E	#001F	#0020
#0021	#0022	#0023	#0024	#0025	#0026	#0027	#0028
#0029	#002A	#002B	#002C	#002D	#002E	#002F	#0030
#0031	#0032	#0033	#0034	#0035	#0036	#0037	#0038
#0039	#003A	#003B	#003C	#003D	#003E	#003F	#0040
#0041	#0042	#0043	#0044	#0045	#0046	#0047	#0048
#0049	#004A	#004B	#004C	#004D	#004E	#004F	#0050
#0051	#0052	#0053	#0054	#0055	#0056	#0057	#0058
#0059	#005A	#005B	#005C	#005D	#005E	#005F	#0060
#0061	#0062	#0063	#0064	#0065	#0066	#0067	#0068
#0069	#006A	#006B	#006C	#006D	#006E	#006F	#0070
#0071	#0072	#0073	#0074	#0075	#0076	#0077	#0078
#0079	#007A	#007B	#007C	#007D	#007E	#007F	#0080
#0081	#0082	#0083	#0084	#0085	#0086	#0087	#0088
#0089	#008A	#008B	#008C	#008D	#008E	#008F	#0090
#0091	#0092	#0093	#0094	#0095	#0096	#0097	#0098
#0099	#009A	#009B	#009C	#009D	#009E	#009F	#00A0
#00A1	#00A2	#00A3	#00A4	#00A5	#00A6	#00A7	#00A8
#00A9	#00AA	#00AB	#00AC	#00AD	#00AE	#00AF	#00B0
#00B1	#00B2	#00B3	#00B4	#00B5	#00B6	#00B7	#00B8
#00B9	#00BA	#00BB	#00BC	#00BD	#00BE	#00BF	#00C0
#00C1	#00C2	#00C3	#00C4	#00C5	#00C6	#00C7	#00C8
#00C9	#00CA	#00CB	#00CC	#00CD	#00CE	#00CF	#00D0
#00D1	#00D2	#00D3	#00D4	#00D5	#00D6	#00D7	#00D8
#00D9	#00DA	#00DB	#00DC	#00DD	#00DE	#00DF	#00E0
#00E1	#00E2	#00E3	#00E4	#00E5	#00E6	#00E7	#00E8
#00E9	#00EA	#00EB	#00EC	#00ED	#00EE	#00EF	#00F0
#00F1	#00F2	#00F3	#00F4	#00F5	#00F6	#00F7	#00F8
#00F9	#00FA	#00FB	#00FC	#00FD	#00FE	#00FF	#0100

【図33】



【図 10】

	情報元	識別子	FLG	TOP	BIN	NUM
SYS	H 00000	—	01	XX--XX	#0005	#00FF 00FB XXXX 分システム情報 XXXX
MF>	H 00020	#0000	01	XX--XX	#0001	#0004 0002 XXXX MF トラック情報 XXXX
R-EF	H 00040	#0001	01	XX--XX	#0002	#0003 0002 XXXX EF トラック情報 XXXX
	H 00060	#0002				
	H 00080	#0003				
DF-EF	H 000A0	#0004	01	XX--XX	FFFF	FFFF XXXX EF トラック情報 XXXX
	H 000C0	#0005	FF		FFFF	FFFF
	H 000E0	#0006	FF		FFFF	FFFF
	H 00100	#0007	FF		FFFF	FFFF
	H 00120	#0008	FF		FFFF	FFFF
	H 00140	#0009	FF		FFFF	FFFF
	H 00160	#000A	FF		FFFF	FFFF
	H 00180	#000B	FF		FFFF	FFFF
	H 001A0	#000C	FF		FFFF	FFFF
	H 001C0	#000D	FF		FFFF	FFFF
	H 001E0	#000E	FF		FFFF	FFFF
	H 00200	#000F	FF		FFFF	FFFF
	H 00220	#0010	FF		FFFF	FFFF
	H 00240	#0011	FF		FFFF	FFFF
	H 00260	#0012	FF		FFFF	FFFF
	H 00280	#0013	FF		FFFF	FFFF
	H 002A0	#0014	FF		FFFF	FFFF
	H 002C0	#0015	FF		FFFF	FFFF
	H 002E0	#0016	FF		FFFF	FFFF
	H 00300	#0017	FF		FFFF	FFFF
:	:	:	FFF			FFF
:	:	:	FFF			FFF
:	:	:	FFF			FFF

H 05000	#0001	#0002	#0003	#0004	#0005	#0006	#0007
	FFF	#0004	#0003	FFF	FFF	#0005	#0007
H 05010	#0006	#0003	#0004	#0005	#0006	#0007	#0008
	#0009	#000A	#000B	#000C	#000D	#000E	#000F
H 05020	#0010	#0011	#0012	#0013	#0014	#0015	#0016
	#0017	#0018	#0019	#001A	#001B	#001C	#001D
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
						#001E	#001F
						#001F	FFF

【图 12】

	制御ディ	データディ	FLG	TOP	ETH	LEN	
SYS	H 000000	—	01	XX...XXXX	#0008	#00FF	00F8 XXXX システムリソース情報 XXXX
MF>	H 000020	#0001	01	XX...XXXX	#0001	#0005	0003 XXXX MF ドラック情報 XXXX
R-FF	H 000040	#0001	01	XX...XXXX	#0002	#0003	0002 XXXX FF ドラック情報 XXXX
	H 000060	#0002					
	H 000080	#0003					
	H 000100	#0004					
DF-FF	H 000120	#0005	01	XX...XXXX	#0006	#0007	0002 XXXX FF ドラック情報 XXXX
DF1	H 000140	#0005	01	XX...XXXX	FFFF	FFFF	FFFF XXXX DF ドラック情報 XXXX
	H 000160	#0006	01	#0005			
	H 000180	#0007					
	H 000200	#0008					
	H 000220	#0009	FF		FFFF	FFFF	FFFF
	H 000240	#000A	FF		FFFF	FFFF	FFFF
	H 000260	#000B	FF		FFFF	FFFF	FFFF
	H 000280	#000C	FF		FFFF	FFFF	FFFF
	H 000300	#000D	FF		FFFF	FFFF	FFFF
	H 000320	#000E	FF		FFFF	FFFF	FFFF
	H 000340	#000F	FF		FFFF	FFFF	FFFF
	H 000360	#0010	FF		FFFF	FFFF	FFFF
	H 000380	#0011	FF		FFFF	FFFF	FFFF
	H 000400	#0012	FF		FFFF	FFFF	FFFF
	H 000420	#0013	FF		FFFF	FFFF	FFFF
	H 000440	#0014	FF		FFFF	FFFF	FFFF
	H 000460	#0015	FF		FFFF	FFFF	FFFF
	H 000480	#0016	FF		FFFF	FFFF	FFFF
	H 000500	#0017	FF		FFFF	FFFF	FFFF
	:	:	FFFF	FFFF
	:	:	FFFF	FFFF
	:	:	FFFF	FFFF

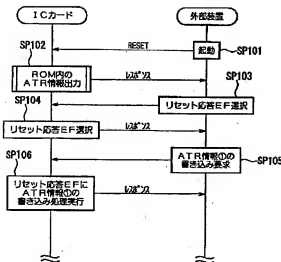
[illegible]

【图 15】

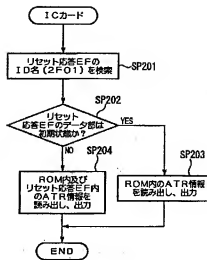
	機器名	装置名	FLG	TOP	BTN	VLM
SFS	H 00000	—	01	XX---XXX	#0014	#00FF 00E0 XXXX 1/4トラック情報 XXXX
MF	H 00002	0001	01	XX---XXX	#0001	#00E 0003 XXXX MFトラック情報 XXXX
R-FF	H 00040	#0001	01	XX---XXX	#0002	#0003 0002 XXXX FFトラック情報 XXXX
	H 00060	#0002	EF 1			
	H 00060	#0003	EF 1			
DF-FF	H 00040	#0004	01	XX---XXX	#0005	#00D 0004 XXXX FFトラック情報 XXXX
DF1	H 00000	#0005	01	XX---XXX	#0008	#00B 0002 XXXX DFトラック情報 XXXX
	H 00050	#0006	01	#0005	DF1のファイル名	
	H 00100	#0007	01	DF1のファイル名		
EF2	H 00120	#0008	01	XX---XXX	#0009	#00A 0002 XXXX FFトラック情報 XXXX
	H 00140	#0009	EF 2			
	H 00160	#000A	EF 2			
DF2>	H 00180	#000B	01	XX---XXX	#000E	#0011 0002 XXXX FFトラック情報 XXXX
	H 00140	#000C	01	#000B	DF2のファイル名	
	H 00100	#000D	01	DF2のファイル名		
EF3	H 00160	#000E	01	XX---XXX	#000F	#0010 0002 XXXX FFトラック情報 XXXX
	H 00200	#000F	EF 3			
	H 00220	#0010	EF 3			
EF4	H 00240	#0011	01	XX---XXX	#0012	#0013 0002 XXXX FFトラック情報 XXXX
	H 00260	#0012	EF 4			
	H 00280	#0013	EF 4			
	H 00240	#0014	FF		FFF	FFF FFF
	H 00200	#0015	FF		FFF	FFF FFF
	H 00260	#0016	FF		FFF	FFF FFF
	H 00300	#0017	FF		FFF	FFF FFF
	:	:	FFF	:	:	:
	:	:	FFF	:	:	:
	:	:	FFF	:	:	:

	#D0C0	*0001	*0002	*0003	*100A	*000E	*0006	*0007
Hr 050000	FFFF	FFFF	<003	FFFF	<005	FFFF	<007	000D
	*0008	*0009	*000A	*00B3	*100C	*00ED	*00EA	*00FA
-Hr 050100	<<008	<*00A	FFFF	FFFF	<0DD	FFFF	<011	<010
	*0010	*00F1	*0012	*0013	*1014	*0015	*0016	*0017
Hr 050500	FFFF	FFFF	<013	FFFF	<015	<016	<017	<018
							*00FE	*00FF
							<0FF	FFFF

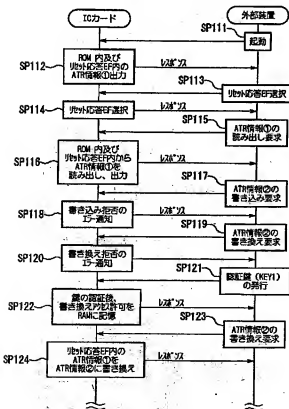
【図 17】



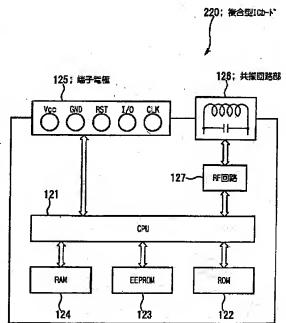
【図 18】



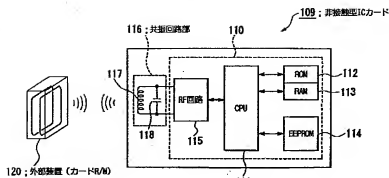
【図 19】



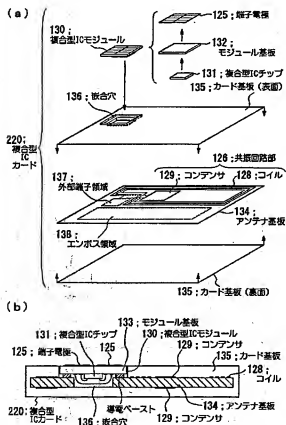
【図 24】



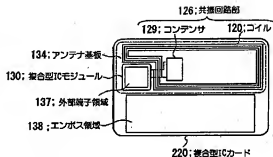
【図 23】



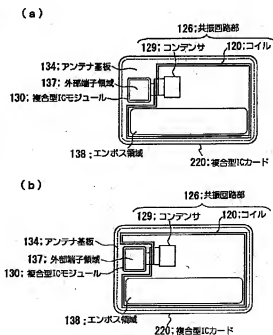
【図 25】



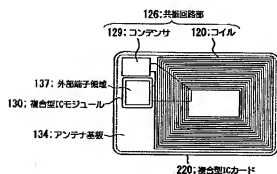
【図 26】



【図 27】



【図28】



【図29】

